

# СОВРЕМЕННЫЕ ТРЕНДЫ ИНФОРМАТИЗАЦИИ. СОЦИО-КИБЕРНЕТИЧЕСКИЕ СИСТЕМЫ

## ПРИМЕНЕНИЕ АВТОМАТНОЙ ПАРАДИГМЫ К СИНТЕЗУ ПРОГРАММЫ УПРАВЛЕНИЯ СТАНКОМ НАПЛАВКИ ПЯТНИКОВ ЖЕЛЕЗНОДОРОЖНЫХ ВАГОНОВ

*Н.А. Афанасьев*

*Научный руководитель: И.А. Тутов, ст. преподаватель ОАР ИШИТР*

*(г. Томск, Томский политехнический университет)*

*e-mail: naa14@tpu.ru*

## APPLICATION AUTOMATON PARADIGM TO A SYNTHESIS OF A MACHINE CONTROL PROGRAM FOR AUTOMATIC SURFACING OF RAILWAY CAR CENTER PLATES

*N.A. Afanasev*

*(Tomsk, Tomsk Polytechnic University)*

**Abstract.** Using an example of a part of the control program for automatic surfacing of railway car center plates, the work demonstrates how the application of an automaton paradigm to the synthesis of programs can allow specialists from different areas of knowledge to clearly understand at what stage the program is currently at, to make the transition from algorithm to program, involving minimum number of control variables, and much more.

**Keywords:** programmable logic controllers, state machine, switch-technology, automaton paradigm, Karnaugh map.

**Введение.** В последнее время объектно-ориентированные языки программирования пользуются большой популярностью среди программистов. Но люди работают не только с персональными компьютерами, в настоящий момент существуют и другие достаточно обширные области, где понятийный аппарат языков такого типа совсем не пригоден. И хотя многие программисты даже не догадываются о существовании этих областей, специалисты насчитывают более двух тысяч разных языков программирования, отличных по специфике, подходу и области использования.

Одна из таких областей – область промышленной автоматизации. В настоящий момент почти все проекты из этой области реализуется на цифровых системах управления, где базовым элементом выступают программируемые логические контроллеры (ПЛК).

Существует международный стандарт [1], который определяет языки программирования для ПЛК. Также есть языки программирования для промышленных, их ещё называют управляющих, компьютеров и для микроконтроллеров [2].

Но при этом, как показал обзор [3], ведущими фирмами в области автоматизации до сих пор не был разработан язык алгоритмизации, который бы дал возможность:

- специалистам из разных областей знаний четко понимать, что им ещё предстоит сделать, а что уже было сделано в программе;
- формально переноситься от алгоритма к программам на различных языках программирования, прибегая к использованию минимального числа внутренних переменных в программах, так чтобы не возникало проблем с пониманием программы;
- легко и верно вносить коррективы в существующие алгоритмы и программный код;

Вопрос создания сквозной технологии алгоритмизации и программирования для систем логического управления, которая позволила бы улучшить качество проектирования их программного обеспечения, оставался открытым из-за отсутствия такого языка.

**Описание конечного автомата.** В [3] было рассмотрено несколько известных технологий программирования и алгоритмизации для таких систем. На основе этого обзора была создана новая технология, которая была названа как "SWITCH-технология", предоставляющая возможность обеспечить исполнение перечисленных выше требований. У этой техноло-

гии есть и другие названия, такие как: "STATE-технология" или "AUTOMATON-технология". Опишем основные аспекты этой технологии, которая, к сожалению, не используется в программистской практике так часто, как она того заслуживает.

Выходное состояние конечного автомата в определенный, рассматриваемый момент времени зависит не только от состояния входов в данный момент времени, но и от внутреннего состояния схемы в момент подачи входных сигналов. При этом, внутреннее состояние схемы зависит от состояния её входов в предшествующий момент времени.

Конечный автомат (КА) целесообразно использовать для структурирования программы. Применение автоматной парадигмы облегчает организацию и сопровождение логики программы, благодаря этому код будет более гибким и надежным [4]. К тому же, КА описывается направленным графом, который понятен специалистам других областей, не специализирующихся на программировании.

В программе может быть несколько конечных автоматов, для каждого из которых определяется свой набор вариаций поведения программы, подлежащих структуризации [5].

**Применение конечного автомата.** Рассмотрим на примере использование конечного автомата при написании программы для системы управления станком наплавки пятников железнодорожных вагонов.

Пусть существуют три режима работы установки: ручной режим (Р), автокалибровка (А) и режим «Наплавка» (Н). Ручной режим доступен после завершения процедуры автокалибровки либо, если она не выполнялась (сигнал «Вкл А» равен лог. «0»), то сразу после подачи питания. Переход из автокалибровки в ручной режим осуществляется при выдаче сигнала «Конец автокалибровки». Для повторного выполнения процедуры автокалибровки необходимо дважды нажать на кнопку «Стоп». Кнопка «Пуск» осуществляет включение установки в режиме «Наплавка». Выход из режима «Наплавка» и переход в режим «Ручной» осуществляется нажатием кнопки «Стоп».

Конечный автомат можно представить в виде направленного графа. Состояния – вершины, а переходы между ними – ребра. На каждом ребре имеется метка, которая информирует о том, при каком условии должен произойти переход. Диаграмма переходов абстрактна в том смысле, что она использует состояния и входы, обозначенные как  $\{X0, X1, X2, X3, X4\}$ . Ниже представлена таблица 1 введенных обозначений.

Таблица 1. Описание переменных

Имя переменной	Описание
X0	Сигнал «Конец автокалибровки»
X1	Нажатие кнопки «Пуск»
X2	Нажатие кнопки «Стоп»
X3	Двойное нажатие кнопки «Стоп»
X4	Активация процедуры автокалибровки

Для построения реальной схемы состояниям должны быть поставлены в соответствие двоичные коды, которые формируются на основе значений выходов, обозначенных как  $\{S0, S1\}$ . КА создает всего одну переменную, хранящую информацию о состоянии программы, а не использует все переменные программы в качестве расширенного определения состояния.

Диаграмма переходов (рисунок 1) в таком случае представляет собой очень наглядную и компактную форму записи алгоритма управления. Для понимания такой формы участникам разработки не нужно обладать специальными знаниями. Кроме того, компактность диаграмма переходов делает подход пригодным для решения задач большой размерности.

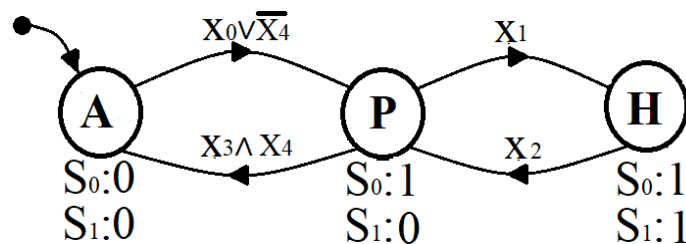


Рисунок 1. Диаграмма переходов

На основе этой диаграммы переходов записана таблица переходов (таблица 2), которая отражает, каким должно быть следующее состояние конечного автомата, соответствующее текущему состоянию и входным сигналам. В таблице переходов используются символ «X» для обозначения безразличных переменных, от которых не зависит состояние выходов.

Таблица 2. Таблица переходов

S <sub>1</sub>	S <sub>0</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	S' <sub>1</sub>	S' <sub>0</sub>
0	0	0	x	x	x	0	0	1
0	0	0	x	x	x	1	0	1
0	0	1	x	x	x	0	0	0
0	0	1	x	x	x	1	0	1
0	1	0	0	x	0	x	0	1
0	1	0	0	x	1	x	1	1
0	1	0	1	x	0	x	0	1
0	1	0	1	x	1	x	1	1
0	1	1	0	x	0	x	0	1
0	1	1	0	x	1	x	1	1
0	1	1	1	x	0	x	0	0
0	1	1	1	x	1	x	1	1
1	1	x	x	0	x	x	1	1
1	1	x	x	1	x	x	0	1

Карта Карно выступает в качестве важнейшего вспомогательным инструментом для того, чтобы определить наиболее простую логическую функцию (таблицы 3-4).

Таблица 3. Карта Карно для сегмента S'<sub>1</sub>

X <sub>3</sub>		0								1							
X <sub>2</sub>		0				1				0				1			
X <sub>1</sub>		0		1		0		1		0		1		0		1	
X <sub>0</sub>		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
S <sub>1</sub> S <sub>0</sub> X <sub>4</sub>	000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	011	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	010	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	110	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
	111	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1

Таблица 4. Карта Карно для сегмента  $S'_0$ 

$X_3$		0								1							
$X_2$		0				1				0				1			
$X_1$		0		1		0		1		0		1		0		1	
$X_0$		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$S_1 S_0 X_4$	000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	001	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	011	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	010	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	111	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Исходя из карт Карно, получим:

$$S'_0 = (S_1 \vee S_0 \vee \bar{x}_4 \vee x_0) \wedge (S_1 \vee \bar{S}_0 \vee \bar{x}_4 \vee \bar{x}_3 \vee x_1);$$

$$S'_1 = \bar{x}_2 \wedge S_0 \wedge S_1 + x_1 \wedge S_0 \wedge \bar{S}_1 = S_0 (\bar{S}_1 \wedge x_1 + S_1 \wedge \bar{x}_2).$$

Для программирования использована программная среда OWEN Logic. Основываясь на полученных выше выражениях, макрос «Конечный автомат выбора режима» примет вид, представленный на рисунке 2.

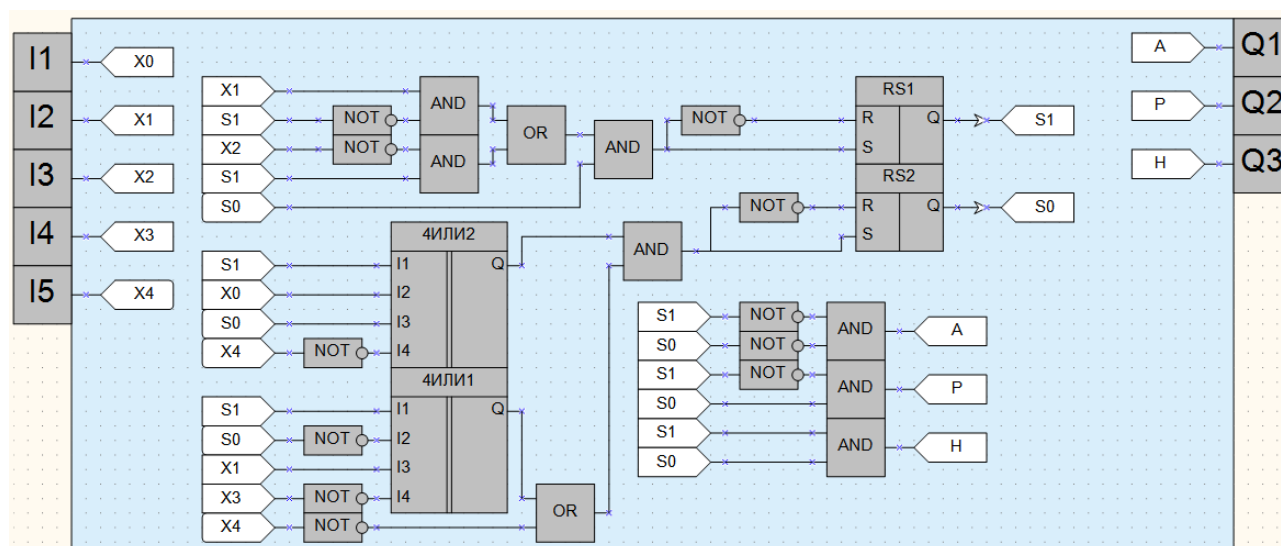


Рисунок 2. Конечный автомат выбора режима работы установки

**Закключение.** Таким образом, на примере было продемонстрировано то, как использование формального описания алгоритма конечным автоматом позволило синтезировать схему на языке FBD или CFC (из стандарта МЭК), в которую могут, в случае необходимости, легко внесены коррективы. Помимо этого, учитывая то, что схема собрана на основе полученных выражений, то она без проблем может быть проверена на наличие ошибок. Кроме того, в полученной программе используется минимальное число внутренних переменных.

#### ЛИТЕРАТУРА

- ГОСТ Р МЭК 61131-3-2016 Контроллеры программируемые. Часть 3. Языки программирования.
- SIMATIC. Simatic S7/M7/C7. Programmable controllers. SIEMENS. Catalog ST 70. 1996.
- Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.

4. Салмре И. Программирование мобильных устройств на платформе .Net Compact Framework. М.: Вильямс. 2006. Кооператив на английском языке — Pearson Education, Inc., 2005.

5. Халимон В.И., Комаров П.И., Рогов А.Ю., Проститенко О.В. Моделирование дискретно-детерминированных процессов с помощью конечных автоматов: учебное пособие/СПбГТИ(ТУ).- СПб, 2007.- 72 с.

## ФОРМИРОВАНИЕ ХРАНИЛИЩА ДАННЫХ ПУТЕЙ ПЕРЕДВИЖЕНИЯ В ГОРОДЕ

<sup>1</sup>А.Т. Ахмедиярова, <sup>2</sup>М.А. Сонькин, <sup>1</sup>В.В. Яворский, <sup>2</sup>О.Б. Фофанов, <sup>3</sup>Е.Г. Ключева, <sup>4</sup>А.О. Чванова

(<sup>1</sup>г. Алматы, Институт информационных и вычислительных технологий,

<sup>2</sup>г. Томск, Томский политехнический университет,

<sup>3</sup>г. Караганда Карагандинский государственный технический университет,

<sup>4</sup>г. Темиртау Карагандинский государственный индустриальный университет)

e-mail: aat.78@mail.ru, sonkin@tpu.ru, yavorskiy-v-v@mail.ru, ofofano@tpu.ru,  
lenchik\_t\_k@mail.ru, mysteria-nastya@mail.ru

## THE FORMATION OF THE DATA WAREHOUSE OF MOVEMENT PATHS IN THE CITY

<sup>1</sup>A.T. Akhmdiyarova, <sup>2</sup>M.A. Sonkin, <sup>1</sup>Yavorskiy V.V., <sup>2</sup>O.B. Fofanov, <sup>3</sup>Ye.G Klyuyeva,  
<sup>4</sup>A.O. Chvanova

(<sup>1</sup>Almaty, Institute of information and computing technologies,

<sup>2</sup>Tomsk, Tomsk Polytechnic University,

<sup>3</sup>Karaganda, Karaganda State Technical University,

<sup>4</sup>Temirtau, Karaganda State Industrial University)

**Abstract.** The paper considers the concept of studying the infrastructure of transport systems. The study of the actual environment is very important for transport systems due to the fact that the actual environment forms and determines the objective needs of infrastructure development and transport technologies. The paper shows that for the rational organization of the transport system it is important to determine the processes of interaction between the control system, the control object and the actual environment. The main product of the current environment for the transport system can be represented in the form of potential correspondence between the districts of the city, points on the transport network, which are then distributed along the route networks. First of all, it is necessary to create data warehouses for the urban public transport management system. The concept of data warehouse involves the implementation of a single integrated data source. As a basis, it is proposed to use a multidimensional model that allows you to allocate different dimensions for different parameters. In the course of the study the models of movement of passengers of urban passenger transport on the basis of Hyper networks are considered. It is shown that the equilibrium state corresponds to the minimum time spent on the movement of passengers and determines the distribution of passengers along the routes in accordance with the strategy of choosing the path. In the proposed model, the equilibrium state determines the distribution of passengers along the routes in accordance with the path selection strategy. The possibilities of modeling the movement of passengers of urban passenger transport on the basis of Hyper networks are considered.

**Keywords:** intelligent transport systems, "smart" city, data storage, movement, urban transport, hypernetwork.

**Введение.** Современные городские транспортные системы (ГТС) – сложные образования, объединяющие инфраструктуру, функциональную структуру и среду потребления транспортных услуг [1].

Инфраструктура – это, в основном, материально-технические средства и системы, необходимые для обеспечения передвижения: улично-дорожные и транспортные сети, обеспечение маршрутных сетей, транспортные предприятия, транспортные средства (подвижной